Math 135: Discrete Mathematics, Fall 2012
Homework 8

Due *in class* on Friday, Nov. 16

---

1. Solve the following *asymptotically*. You need to argue why your bounds are correct (either by induction or using summations and recursion trees or using Master theorem). However, you do not need to prove that your big-O bounds are formally correct; for example, if you get a bound of $n/2 \log n + 12$, you may just say that it is $O(n \log n)$.

   For each, assume small constant base cases, such as $A(1) = O(1)$ or $A(2) = O(1)$.

   (a) $A(n) = 5A(n/5) + n^3$

   (b) $B(n) = 4B(n/4) + \sqrt{n}$

   (c) $C(n) = 9C(n/2) + (3n^2 + 2)$

   (d) $D(n) = 3D(n/3) + 2n - 3$

   (e) $E(n) = 2E(\sqrt{n}) + 1$

   (f) $F(n) = F(n/2) + F(n/4) + F(n/6) + F(n/12) + n$

2. Analyze the runtime of the following algorithms. ("Analyze the runtime" means give a simple function $f(n)$ such that the runtime of the algorithm is $\Theta(f(n))$.) You should explain how you got your answers, but you do not need to given formal proofs.

(a)
```
ALGA(n):
  if n = 0 or n = 1 then
      return 1
  else
      x ← ALGA(n − 1)
      y ← ALGA(n − 2)
      return 2x + 3y
```

(b)
```
ALGB(A[1...n]):
  if n ≤ 1 then
      return 0
  else
      k ← ⌊n/2⌋
      x ← ALGB(A[1...k])
      y ← ALGB(A[k+1...n])
      z ← 0
      for i = 1 to k
          for j = k+1 to n
              z ← z + A[i] * A[j]
      return x + y + z
```

(c)
```
ALGC(A[1...n]):
  i ← ⌈n/3⌉ + 1
  j ← ⌊2n/3⌋
  if n = 1 then
      return A[1]
  else
      x ← ALGC(A[1...j])
      y ← ALGC(A[i...n])
      z ← 0 for k = i to j do
          z ← z + A[k]
      return xy + z
```